**EXP:01**

**Step by step procedure for HADOOP installation on UBUNTU**

**Procedure:**

Open terminal in UBUNTU and execute the fallowing commands one by one

 1)sudo apt update  
  
2)sudo apt install openjdk-8-jdk -y  
  
3)java -version; javac -version  
  
4)sudo apt install openssh-server openssh-client -y  
  
5)ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa  
  
6)cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  
  
7)chmod 0600 ~/.ssh/authorized\_keys  
  
8)ssh localhost  
  
9)wget https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz  
  
10)tar xzf hadoop-3.2.1.tar.gz  
  
11)sudo nano .bashrc  
  
#Hadoop Related Options  
export HADOOP\_HOME=/home/hadoop-3.2.1  
export HADOOP\_INSTALL=$HADOOP\_HOME  
export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME  
export HADOOP\_COMMON\_HOME=$HADOOP\_HOME  
export HADOOP\_HDFS\_HOME=$HADOOP\_HOME  
export YARN\_HOME=$HADOOP\_HOME  
export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native  
export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin  
export HADOOP\_OPTS=”-Djava.library.path=$HADOOP\_HOME/lib/native”  
  
12)source ~/.bashrc  
  
13)sudo nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh  
  
export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
  
which javac  
  
readlink -f /usr/bin/javac  
  
  
14)sudo nano $HADOOP\_HOME/etc/hadoop/core-site.xml  
  
<configuration>  
<property>  
  <name>hadoop.tmp.dir</name>  
  <value>/home/tmpdata</value>  
</property>  
<property>  
  <name>fs.default.name</name>  
  <value>hdfs://127.0.0.1:9000</value>  
</property>  
</configuration>  
  
15)sudo nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml  
  
<configuration>  
<property>  
  <name>dfs.data.dir</name>  
  <value>/home/dfsdata/namenode</value>  
</property>  
<property>  
  <name>dfs.data.dir</name>  
  <value>/home/dfsdata/datanode</value>  
</property>  
<property>  
  <name>dfs.replication</name>  
  <value>1</value>  
</property>  
</configuration>  
  
16)sudo nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml  
  
<configuration>  
<property>  
  <name>mapreduce.framework.name</name>  
  <value>yarn</value>  
</property>  
</configuration>  
  
  
17)sudo nano $HADOOP\_HOME/etc/hadoop/yarn-site.xml  
<configuration>  
<property>  
  <name>yarn.nodemanager.aux-services</name>  
  <value>mapreduce\_shuffle</value>  
</property>  
<property>  
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>  
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>  
</property>  
<property>  
  <name>yarn.resourcemanager.hostname</name>  
  <value>127.0.0.1</value>  
</property>  
<property>  
  <name>yarn.acl.enable</name>  
  <value>0</value>  
</property>  
<property>  
  <name>yarn.nodemanager.env-whitelist</name>     
  <value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PERPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>  
</property>  
</configuration>  
  
18)hdfs namenode -format  
  
Navigate to the hadoop-3.2.1/sbin directory  
  
19)./start-dfs.sh  
  
20)./start-yarn.sh  
  
21)jps  
  
22)http://localhost:9870  
  
23)http://localhost:9864  
  
24)http://localhost:8088

**Refreshing UNIX Commands**

# 1. Print the Hadoop version  
hadoop version

# 2. List the contents of the root directory in HDFS  
#  
hadoop fs -ls /

# 3. Report the amount of space used and  
# available on currently mounted filesystem  
#  
hadoop fs -df hdfs:/

# 4. Count the number of directories,files and bytes under  
# the paths that match the specified file pattern  
#  
hadoop fs -count hdfs:/

# 5. Run a DFS filesystem checking utility  
#  
hadoop fsck – /

# 6. Run a cluster balancing utility  
#  
hadoop balancer

# 7. Create a new directory named “hadoop” below the  
# /user/training directory in HDFS. Since you’re  
# currently logged in with the “training” user ID,  
# /user/training is your home directory in HDFS.  
#  
hadoop fs -mkdir /user/training/hadoop

# 8. Add a sample text file from the local directory  
# named “data” to the new directory you created in HDFS  
# during the previous step.  
#  
hadoop fs -put data/sample.txt /user/training/hadoop

# 9. List the contents of this new directory in HDFS.  
#  
hadoop fs -ls /user/training/hadoop

# 10. Add the entire local directory called “retail” to the  
# /user/training directory in HDFS.  
#  
hadoop fs -put data/retail /user/training/hadoop

# 11. Since /user/training is your home directory in HDFS,  
# any command that does not have an absolute path is  
# interpreted as relative to that directory. The next  
# command will therefore list your home directory, and  
# should show the items you’ve just added there.  
#  
hadoop fs -ls

# 12. See how much space this directory occupies in HDFS.  
#  
hadoop fs -du -s -h hadoop/retail

# 13. Delete a file ‘customers’ from the “retail” directory.  
#  
hadoop fs -rm hadoop/retail/customers

# 14. Ensure this file is no longer in HDFS.  
#  
hadoop fs -ls hadoop/retail/customers

# 15. Delete all files from the “retail” directory using a wildcard.  
#  
hadoop fs -rm hadoop/retail/\*

# 16. To empty the trash  
#  
hadoop fs -expunge

# 17. Finally, remove the entire retail directory and all  
# of its contents in HDFS.  
#  
hadoop fs -rm -r hadoop/retail

# 18. List the hadoop directory again  
#  
hadoop fs -ls hadoop

# 19. Add the purchases.txt file from the local directory  
# named “/home/training/” to the hadoop directory you created in HDFS  
#  
hadoop fs -copyFromLocal /home/training/purchases.txt hadoop/

# 20. To view the contents of your text file purchases.txt  
# which is present in your hadoop directory.  
#  
hadoop fs -cat hadoop/purchases.txt

# 21. Add the purchases.txt file from “hadoop” directory which is present in HDFS directory  
# to the directory “data” which is present in your local directory  
#  
hadoop fs -copyToLocal hadoop/purchases.txt /home/training/data

# 22. cp is used to copy files between directories present in HDFS  
#  
hadoop fs -cp /user/training/\*.txt /user/training/hadoop

# 23. ‘-get’ command can be used alternaively to ‘-copyToLocal’ command  
#  
hadoop fs -get hadoop/sample.txt /home/training/

# 24. Display last kilobyte of the file “purchases.txt” to stdout.  
#  
hadoop fs -tail hadoop/purchases.txt

# 25. Default file permissions are 666 in HDFS  
# Use ‘-chmod’ command to change permissions of a file  
#  
hadoop fs -ls hadoop/purchases.txt  
sudo -u hdfs hadoop fs -chmod 600 hadoop/purchases.txt

# 26. Default names of owner and group are training,training  
# Use ‘-chown’ to change owner name and group name simultaneously  
#  
hadoop fs -ls hadoop/purchases.txt  
sudo -u hdfs hadoop fs -chown root:root hadoop/purchases.txt

# 27. Default name of group is training  
# Use ‘-chgrp’ command to change group name  
#  
hadoop fs -ls hadoop/purchases.txt  
sudo -u hdfs hadoop fs -chgrp training hadoop/purchases.txt

# 28. Move a directory from one location to other  
#  
hadoop fs -mv hadoop apache\_hadoop

# 29. Default replication factor to a file is 3.  
# Use ‘-setrep’ command to change replication factor of a file  
#  
hadoop fs -setrep -w 2 apache\_hadoop/sample.txt

# 30. Copy a directory from one node in the cluster to another  
# Use ‘-distcp’ command to copy,  
# -overwrite option to overwrite in an existing files  
# -update command to synchronize both directories  
#  
hadoop fs -distcp hdfs://namenodeA/apache\_hadoop hdfs://namenodeB/hadoop

# 31. Command to make the name node leave safe mode  
#  
hadoop fs -expunge  
sudo -u hdfs hdfs dfsadmin -safemode leave

# 32. List all the hadoop file system shell commands  
#  
hadoop fs

# 33. Last but not least, always ask for help!  
#  
hadoop fs -help